PERBANDINGAN CAS DAN OAUTH DALAM SINGLE SIGN ON (SSO)

Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Jurusan Informatika Fakultas Komunikasi dan Informatika

Oleh:

IMAM ANSHORUDIN

L 200 120 129

PROGRAM STUDI INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2016

HALAMAN PERSETUJUAN

PERBANDINGAN CAS DAN OAUTH DALAM SINGLE SIGN ON (SSO)

PUBLIKASI ILMIAH

oleh:

IMAM ANSHORUDIN

L 200 120 129

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing

Helman Muhammad, S.T., M.T.

NIK.1564

HALAMAN PENGESAHAN

PERBANDINGAN CAS DAN OAUTH DALAM SINGLE SIGN ON (SSO)

OLEH

IMAM ANSHORUDIN

L 200 120 129

Telah dipertahankan di depan Dewan Penguji

Fakultas Komunikasi Dan Informatika

Universitas Muhammadiyah Surakarta

Pada hari Sabtu, 22 Oktober **2016**

dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. **Helman Muhammad, S.T., M.T.** (.....)
   (Ketua Dewan Penguji)
2. **Husni Thamrin, S.T., M.T., Ph.D.** (.....)
   (Anggota I Dewan Penguji)
3. **Dr. Heru Supriyono, M.Sc.** (.....)
   (Anggota II Dewan Penguji)

Publikasi ilmiah ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar sarjana

Tanggal 3 November 2016

Mengetahui,

Dekan

Fakultas Komunikasi dan Informatika

Husni Thamrin, S.T., M.T., Ph.D.

NIK : 706

Ketua Program Studi

Informatika

Dr. Heru Supriyono, M.Sc.

NIK:970

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam naskah publikasi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

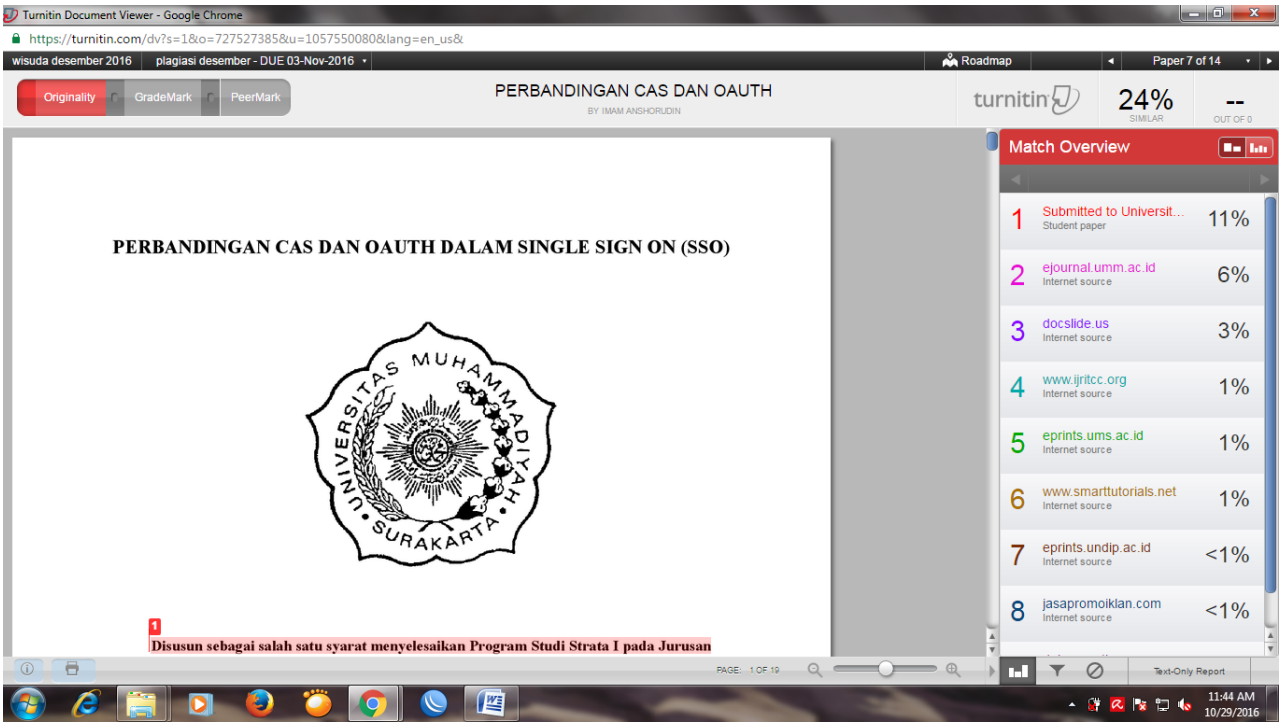
Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 1 November 2016

Penulis

IMAM ANSHORUDIN

L 200 120 129


Turnitin Document Viewer - Google Chrome
https://turnitin.com/dv?s=1&o=727527385&u=1057550080&lang=en_us&
wisuda desember 2016
plagiasi desember - DUE 03-Nov-2016
Roadmap
Paper 7 of 14
Originality
GradeMark
PeerMark
PERBANDINGAN CAS DAN OAUTH
BY IMAM ANSHORUDIN
turnitin
24%
SIMILAR
--
OUT OF 0
PERBANDINGAN CAS DAN OAUTH DALAM SINGLE SIGN ON (SSO)
UNIVERSITAS MUHAMMADIYAH SURAKARTA
Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Jurusan
Match Overview
1 Submitted to Universit... Student paper 11%
2 ejournal.umm.ac.id Internet source 6%
3 docslide.us Internet source 3%
4 www.ijritcc.org Internet source 1%
5 eprints.ums.ac.id Internet source 1%
6 www.smarttutorials.net Internet source 1%
7 eprints.undip.ac.id Internet source <1%
8 jasapromoiklan.com Internet source <1%
PAGE: 1 OF 19
Text-Only Report
11:44 AM
10/29/2016

UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA
Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: http://informatika.ums.ac.id. Email: informatika@ums.ac.id

SURAT KETERANGAN LULUS PLAGIASI

012/A.3-II.3/INF-FKI/I/2016

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama	:	IMAM ANSHORUDIN
NIM	:	L200120129
Judul	:	PERBANDINGAN CAS DAN OAUTH DALAM SINGLE SIGN ON (SSO)
Program Studi	:	Informatika
Status	:	**Lulus**

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 28 Oktober 2016

Biro Skripsi Informatika

Ihsan Cahyo Utomo, S.Kom., M.Kom.

PERBANDINGAN CAS DAN OAUTH DALAM SINGLE SIGN ON (SSO)

ABSTRAK

Single Sign On merupakan salah satu sistem yang telah dikembangkan sejak dulu untuk memenuhi harapan para pengembang dan memberi kemudahan serta kenyamanan dalam hal mengakses data. Dalam perkembangannya, terbentuklah metode-metode dan protokol-protokol yang bervariasi untuk menyesuaikan dengan kebutuhan para pengembang. Dalam bermacam-macam metode dan protokol, seorang pengembang dapat memilih architecture dan protokol yang dapat digunakan untuk mengembangkan sistemnya. Central Authentication Service dan Open Autorization merupakan dua sistem Single Sign On yang paling banyak digunakan dalam pembuatan web login. Keduanya dapat dijadikan sebagai dasar untuk penerapan sistem Single Sign On bagi pengembang yang berniat merancang sistem login yang aman dan nyaman, sehingga pengembang dapat menciptakan sistem yang sesuai dengan keinginanya.

Kata Kunci: Central Authentication Service, Open Authorization, Tomcat, Single Sign On, Php

ABSTRACT

Single Sign On is one of the systems that have been developed long ago to meet the expectations of developers to provide ease and convenience of accessing data. In the development of the system, methods and protocols have been forming in varied way to suit the needs of the developers . In a variety of methods and protocols , a developer can choose the architecture and protocols that can be used to develop the system. Central Authentication Service and Open autorization is two Single Sign On systems most widely used in the manufacture of a web log . Both can be used as the basis for the application of the system of Single Sign On for developers who intend to design a login system that is safe and comfortable , so that developers can create a system that suits his desire .

Keywords: Central Authentication Service, Open Authorization, Tomcat, Single Sign On, Php

1.PENDAHULUAN

Web merupakan media interface yang dibuat sedemikian rupa untuk memberikan kenyamanan dan kemudahan dalam mengakses informasi-informasi yang tersedia di dalam internet. Perkembangan suatu web tidak terlepas dari perkembangan sistem-sistem yang ada didalamnya, salah satunya adalah sistem login yang kini mulai menjadi perhatian bagi para pengembang web. Dimana kenyamanan dan keamanan web diciptakan seiring dengan perkembangan web itu sendiri. Pemberian hak akses adalah dasar dari sistem login, dimana setiap pengguna bisa atau tidak untuk mengakses suatu informasi itu tergantung pada hak yang diterimanya. (Al-Fedaghi 2011).

Single Sign On merupakan sistem yang diciptakan untuk memudahkan para pengguna, dimana pengguna hanya perlu melakukan login sekali saja agar dapat mengkases seluruh aplikasi yang telah terintegrasi dengan protokol SSO. Sehingga para pengembang beromba-lomba dalam memaskimalkan sistem login pada web yang mereka kembangkan, dimana banyak metode yang dapat digunakan dan protokol-protokol yang dapat di implementasikan. (Grag 2016.)

Aminudin (2014) dalam skripsinya yang berjudul "Implementation of Single Sign On (SSO) Support For E-Commerce Interactivity Applications Using Protocol Oauth" menyatakan bahwa dengan menggunakan SSO, pengguna hanya cukup berusaha untuk otentikasi hanya sekali untuk mendapatkan izin, akses ke semua layanan yang terdapat dalam jaringan.Dengan menggunakan protokol OAuth, pengguna dapat mengotorisasi klien untuk mengakses data yang dilindungi sudah berada di server dengan memberikan token tanpa mengajukan username dan password. OAuth memungkinkan pengguna untuk memberikan akses ke situs pihak ketiga untuk mengakses informasi yang tersimpan pada penyedia layanan lain tanpa harus membagi hak akses atau semua data mereka. Single Sign On sistem dengan protokol OAuth digunakan adalah teknologi otentikasi dengan kode tanda bukan username dan password. Penelitian ini diharapkan dapat memudahkan pengguna untuk mengotentikasi aplikasi e-commerce dengan menggunakan penyedia account yang mendukung protokol OAuth, sehingga efek positif pada perdagangan.

Ramadhan (2014) dalam skripsinya yang berjudul "Analisis teknologi Single Sign On (SSO) dengan penerapan Central Authentication Service (CAS) pada Universitas Bina Darma" menyatakan bahwa CAS merupakan protocol SSO yang bertujuan memberikan ijin pada pengguna dalam mengakses beberapa aplikasi, sekaligus menyediakan credential pengguna (seperti user id dan password) dalam sekali login, dan mengizinkan aplikasi web untuk meng-otentikasi pengguna tanpa mendapatkan akses ke security credential pengguna,

hal ini dapat mempermudah pengguna dalam menggunakan aplikasi yang ada dan juga dapat mempermudah dalam pengorganisasian data pengguna, sehingga keamanan data pengguna lebih terjamin.

Penelitian ini ditujukan untuk menjadi acuan atau referensi bagi para pengembang agar memudahkan dalam implementasi pengembangan SSO, dalam penelitian ini akan dibandingkan antara CAS dan OAuth dimana kedua sistem login ini diambil sebagai dasar dari SSO. Pembuatan sistem web login CAS dan OAuth berjalan pada Apache dan Tomcat, sedangkan untuk test performa menggunakan Apache Bench.

2.METODE

Dari penelitian ini akan diketahui kemudahan perancangan web login antara CAS dan OAuth sebagai dasar SSO yang mengacu pada jurnal, artikel, thesis dan tutorial yang memudahkan dalam pembuatan sistem login, serta bagaimana proses dari sistem login central authentication service dan open authorization sebagai sistem login dalam SSO. Pembuatan web login ini berdasarkan pada kebutuhan dari dua sistem yang akan dijadikan sebagai contoh, dimana default konfigurasi untuk cas dan simple web login dengan google account untuk oauth.

Test performa sistem login anatara CAS dan OAuth menggunakan Apache Benc, test yang dilakukan adalah *Response Time,* dimana dalam test ini akan diketahui performa kedua sistem login dalam menangani *request* per satuan detik. Tools Apache Benc dapat dibuka lewat cmd, setelah masuk ke path C:\xampp\apache\bin, maka perintah yang dimasukkan adalah >ab –t [waktu (s)] [http://]hostname[:port]/path. Disini hasil yang dapat diamati adalah jumlah request yang dapat ditangani oleh sistem dalam selang waktu yang telah ditentukan.

2.1 ALAT DAN BAHAN

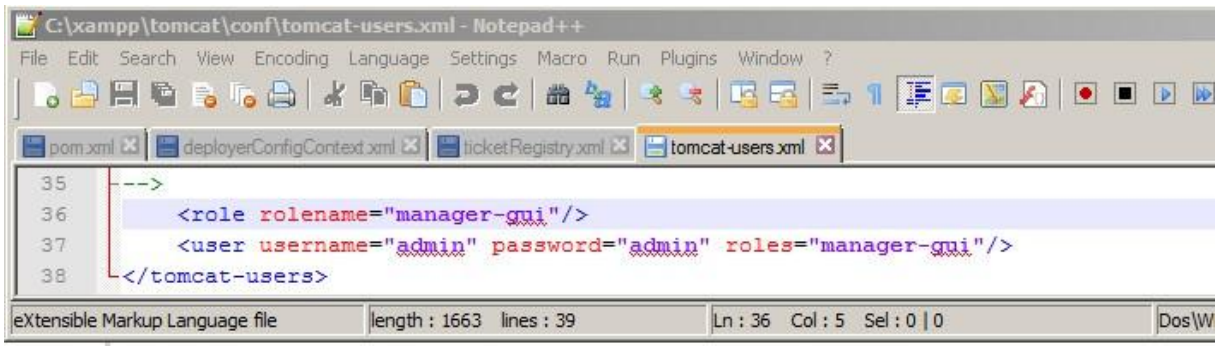
Peralatan utama dalam penelitian ini dibagi menjadi dua kategori yaitu perangkat keras (*hardware)* dan perangkat lunak (*software*). *Hardware* yang digunakan adalah Laptop *Acer ASPIRE E1-471* dengan sistem operasi *Windows 7*dan spesifikasi *Processor Intel Core i3-2328M up to 2,2Ghz, Harddisk 500gB* dan *RAM 2GB*. Software yang digunakan untuk penelitian adalah *Apache, PHP, Mysql,Tomcat 7.0.56, Mozilla Firefox*.

2.2 PERANCANGAN WEB LOGIN CAS DAN OAUTH

Perangkat keras yang digunakan adalah Laptop *Acer ASPIRE E1-471*, sedangkan untuk pembuatan web dilakukan dengan xampp apcahe dan tomcat. Dimana untuk OAuth akan menggunakan php dan mysql untuk pembuatan web login, sedangkan untuk CAS akan menggunakan tomcat dengan default sistem.

2.2.1 Pembuatan cas web login

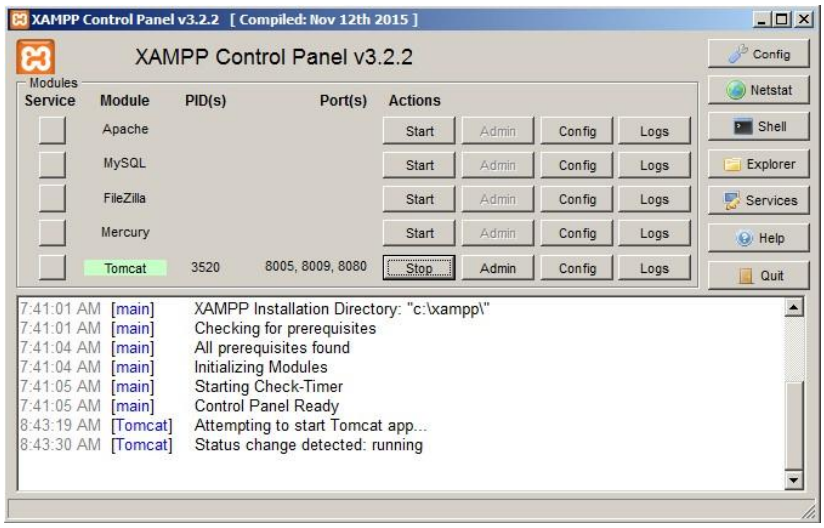
Mengkonfigurasi Tomcat agar dapat mengakses Tomcat manager page, dengan merubah manager-gui dari setingan defaultnya. Untuk dapat mengakses manager page dilakukan penambahan user secara manual pada manager-gui. Edit file "tomcat-users.xml" yang berada di direktori C:\xampp\tomcat\conf, dengan menambahkan username dan password baru.



Gambar 1. Pembuatan User untuk Tomcat

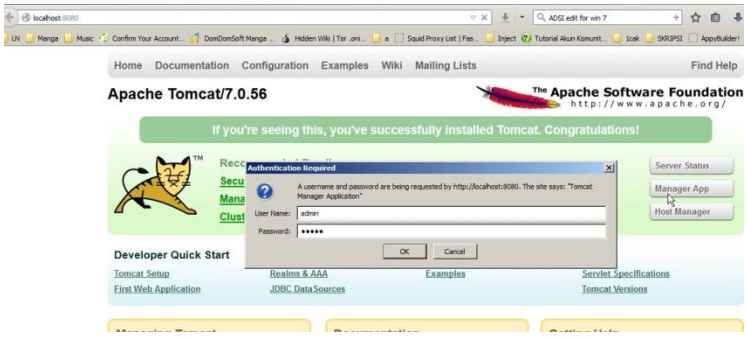
2.2.2 Menambahkan cas-server-webapp

Pada direktori "C:\xampp\tomcat\webapps" tambahkan cas web aplikasi yang tersedia di *http://mvnrepository.com*. Masuk ke tomcat manager page dengan mengaktifkannya dari xampp control panel.



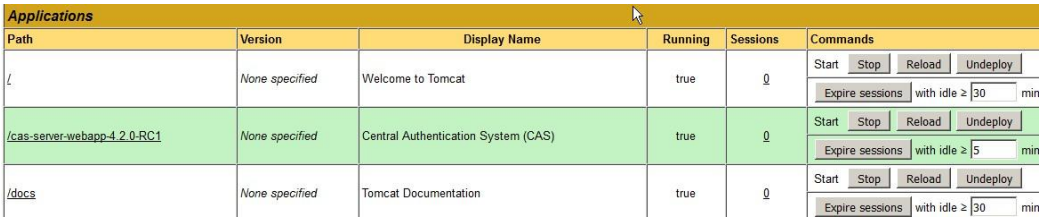
Gambar 2. Memulai Tomcat

Setelah muncul tampilan manager page dari tomcat, muncul permintaan autentikasi username dan password dimana username dan password tersebut di isi dengan admin, sesuai dengan penambahan user pada "tomcat-users.xml".



Gambar 3. Masuk ke Manager App Tomcat

Di dalam Tomcat Web Application Manager pada kolom Application sudah terdapat path dari "cas-server-webapp-4.2.0-RC1" yang merupakan webapp cas yang masih default.



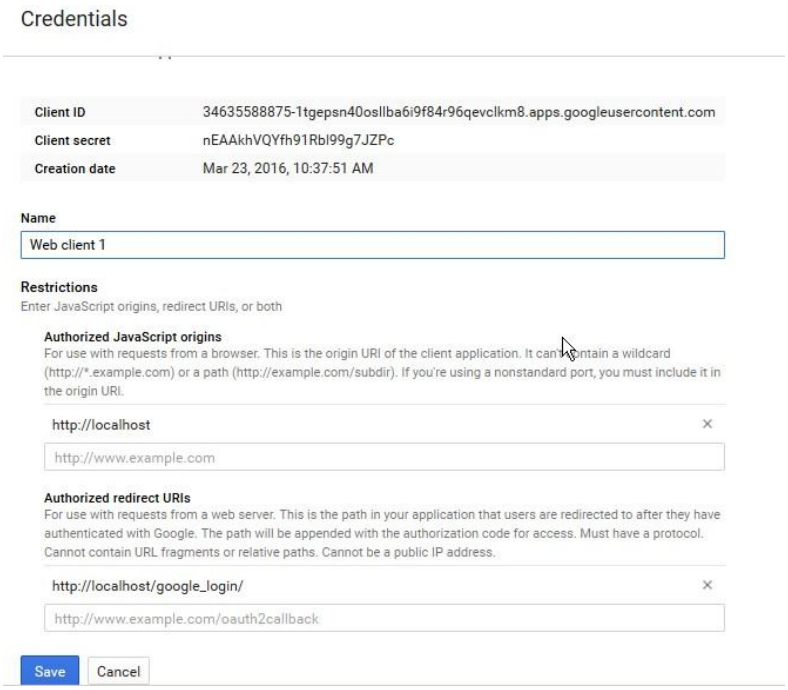
Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 min
/cas-server-webapp-4.2.0-RC1	*None specified*	Central Authentication System (CAS)	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 5 min
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 min

Gambar 4. Tampilan Application Manager pada Tomcat

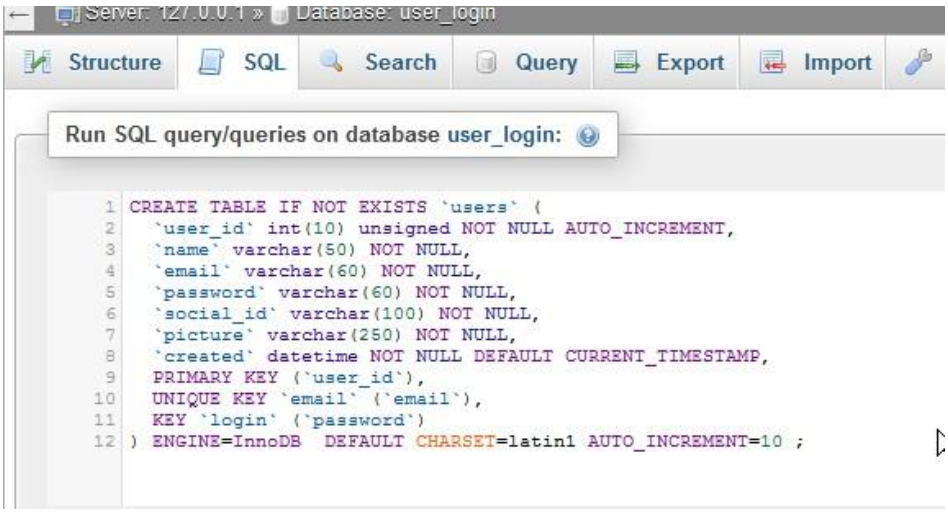
2.2.3 Pembuatan OAuth web login

Membuat Google App untuk mendapatkan Google Oauth Client ID dan Client Secret dari https://code.google.com/apis/console. Set path Authorized JavaScript origins dengan "http://localhost" dan set Authorized redirect URIs dengan http://localhost/google_login/.



Gambar 5. Pembuatan Google Client ID, Client Secret, dan mensetting path URIs

Membuat database “user_login” dengan table “users” dengan SQL query.



Gambar 6. Pembuatan Database OAuth

Membuat “config.php” dan set Client ID dan Client Secret yang telah di dapat dari Google App.

Gambar 7. Konfigurasi config.php

Membuat “index.php”, dimana ketika google+ button di klik maka OAuth API akan meminta request ke google.

```
<h4 class="text-center login-txt-center">Log in alternatif:</h4>

<a class="btn btn-default google" href="<?php echo $client->createAuthUrl();?>">
<i class="fa fa-google-plus modal-icons"></i> Sign In with Google+ </a>
```

Gambar 8. Script Tombol Sign with Google+

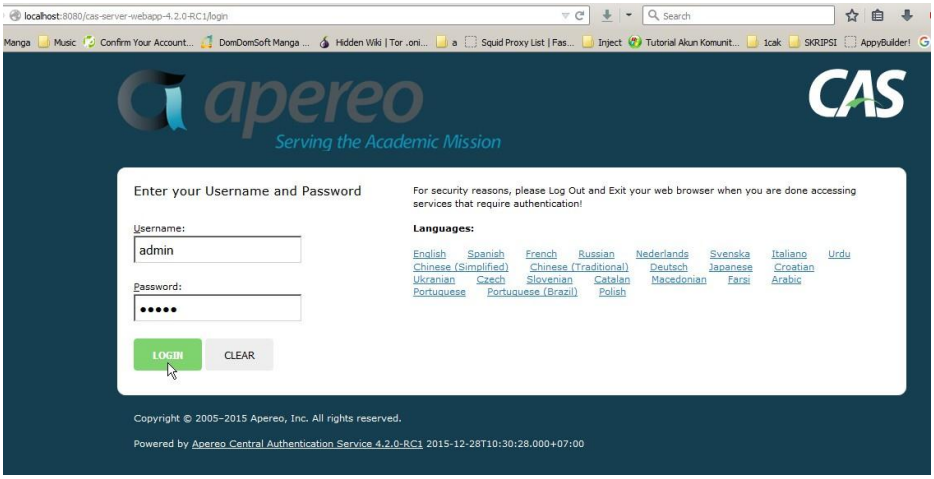
3. HASIL DAN PEMBAHASAN

3.1 HASIL PENELITIAN

Setelah pembuatan web login Central Authentication Service dan Open Authorization selesai, maka dilakukan percobaan untuk login dengan username dan password secara default untuk Central Authentication Service dan login dengan google account untuk Open Authorization.

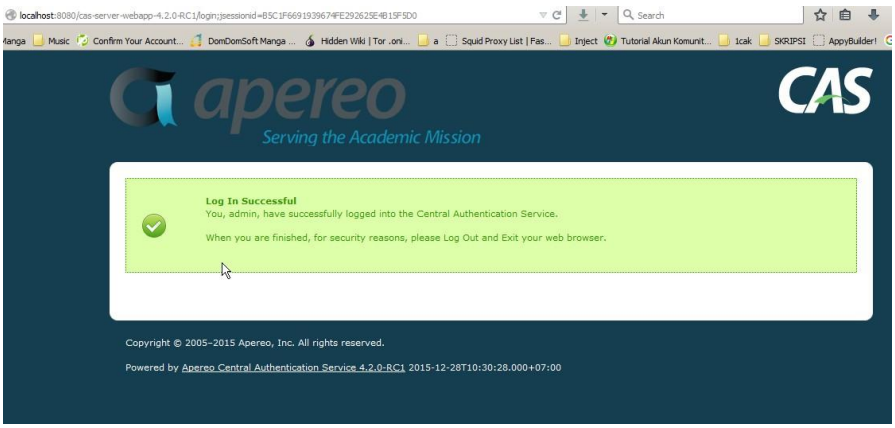
3.1.1 Login kedalam cas web login

Mengakses cas web login dengan memaskukkan alamat http://localhost:8080/cas-server-webapp-4.2.0-RC1/login, dimana username dan password defaultnya adalah casuser dan Mellon. Dengan penambahan username dan password secara langsung melalui "cas.propertis" dengan username:admin dan password:admin.



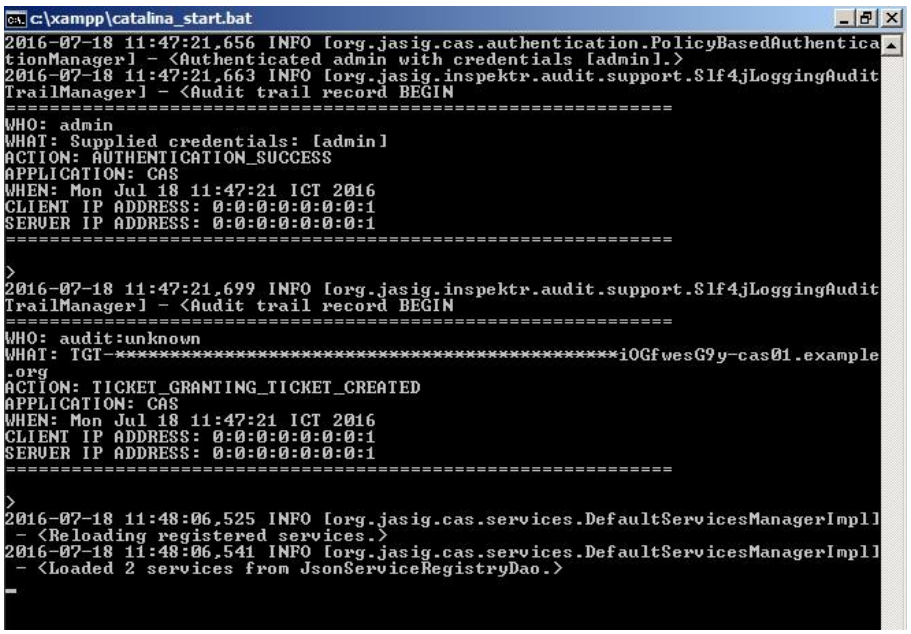
Gambar 9. Tampilan Login Page CAS

Setelah login sukses maka akan muncul tampilan seperti berikut, dan akan muncul peringatan "Log In Successful You, admin, have successfully logged into the Central Authentication Service. When you are finished, for security reasons, please Log Out and Exit your web browser."



Gambar 10. Tampilan Page Setelah Login

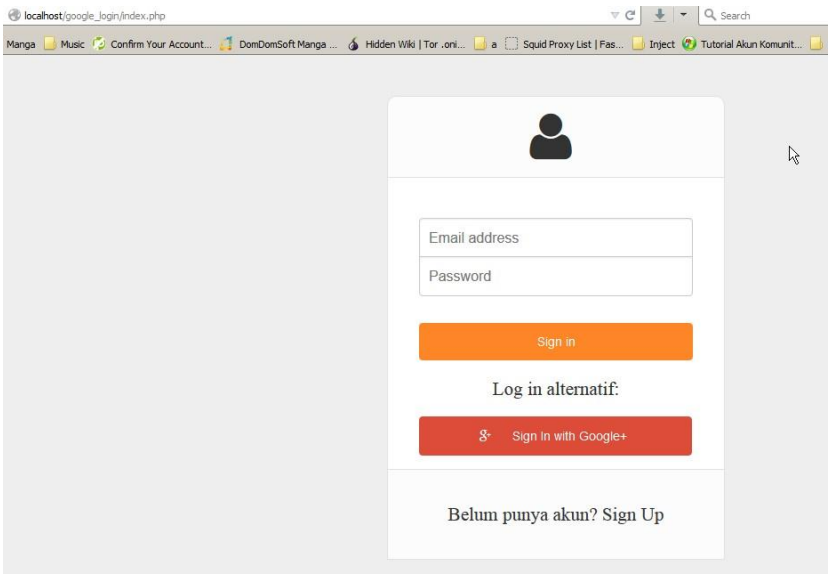
Pada saat yang sama dalam log catalina ter-record pembuatan TGT untuk user admin dimana saat user keluar dari web yang terintegrasi dengan cas maka ticket session yang telah diberikan akan berakhir dan user harus kembali login.



Gambar 11. Informasi Pembuatan TGT dan Authentication Succes dalam catalina.bat

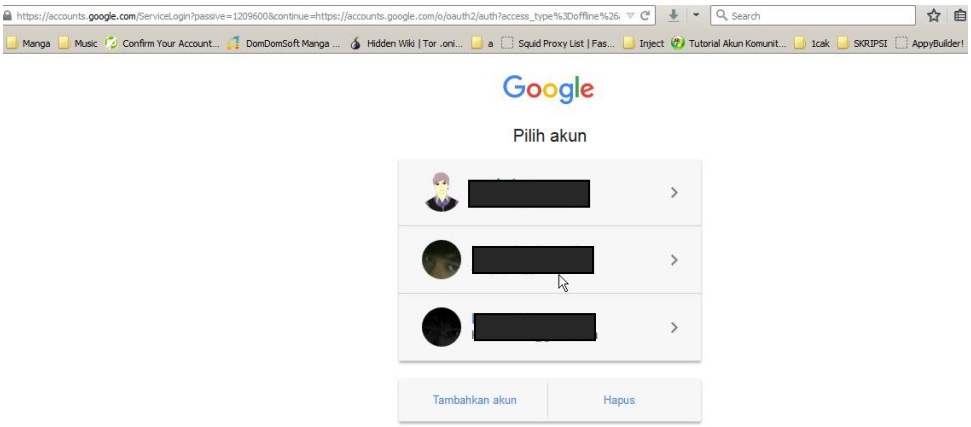
3.1.2 Login ke dalam oauth web login

Akses alamat localhost/google_login, maka akan muncul tampilan login form sederhana dimana terdapat dua field untuk email address dan password. Satu tombol Sign in dan satu tombol Sign in with Google+. Tombol Sign in merupakan tombol login biasa, sedangkan tombol Sign in with Google+ sudah terintegrasi dengan sistem oauth sehingga user akan langsung di forward ke akun google pengguna.



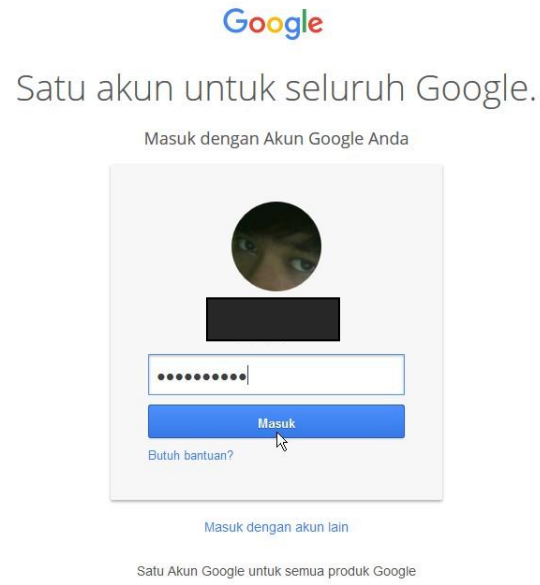
Gambar 12. Tampilan Login Page dari OAuth

Akun google yang akan tampil setelah tombol Sign in with Google+ di klik, menamilkan akun-akun yang dimiliki user. Pilih satu user dan klik untuk memasukkan password.



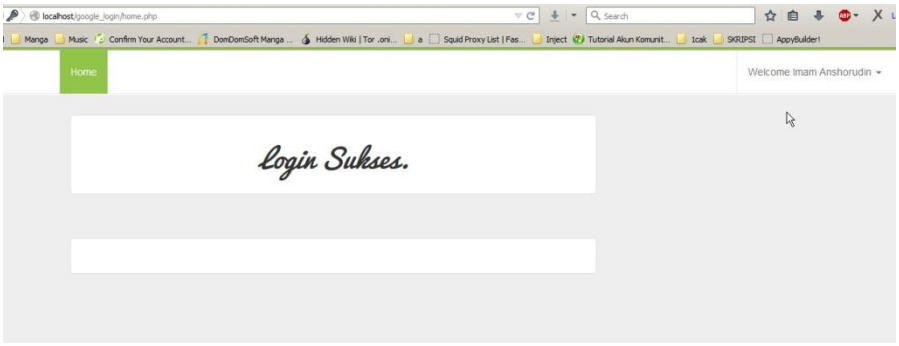
Gambar 13. Tampilan Akun Google yang Tersedia

Akan muncul page yang meminta password akun google si user, jika user tersebut login dengan akun googlenya otomatis dia akan diberikan akses dari web yang ingin di akses.



Gambar 14. Login Akun Google

Halaman beranda/ home setelah login, jika user tidak logout dari akun google-nya maka secara otomatis saat user meengakses alamat localhost/google_login maka akan secara langsung ter direct ke halman home.



Gambar 15. Tampilan Page Setelah Login

3.1.3 Test Performa dengan Apache Benchmark

Test performa yang dilakukan untuk mengetest sistem login CAS dan OAuth adalah menggunakan *Apache Bench* (AB) tools. Tools Apache Benc dapat dibuka lewat cmd, setelah masuk ke path C:\xampp\apache\bin, maka perintah yang dimasukkan adalah:

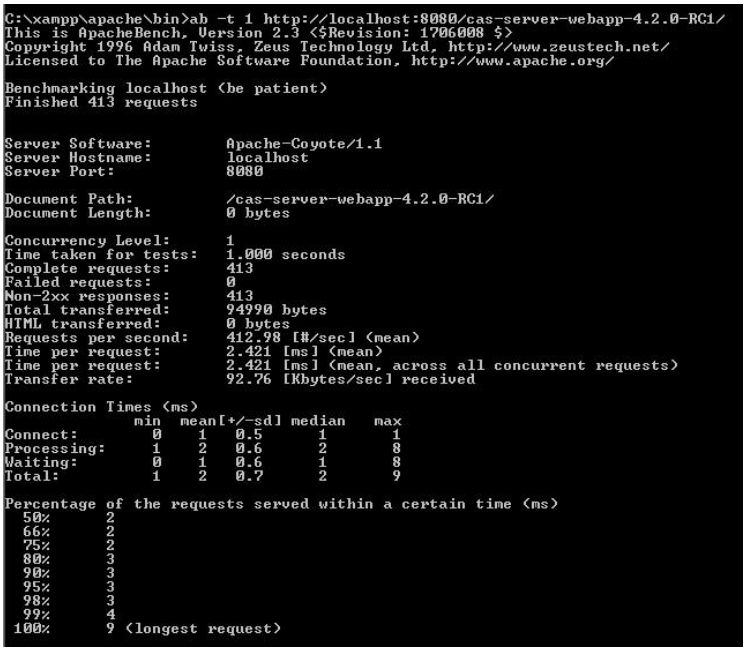
>ab –t 1 localhost/google_login/ (untuk test performa sistem oauth)

>ab –t 1 http://localhost:8080/cas-server-webapp-4.2.0-RC1/login (untuk test performa sistem cas). Disini hasil yang dapat diamati adalah jumlah request yang dapat ditangani oleh sistem dalam selang waktu yang telah ditentukan. Dengan 5 kali percobaan dari selang waktu yang telah ditentukan, diambil rata-rata dari hasil yang didapat. Hasil dari test performa antara sistem login CAS dan OAuth dapat dilihat pada Table 1.

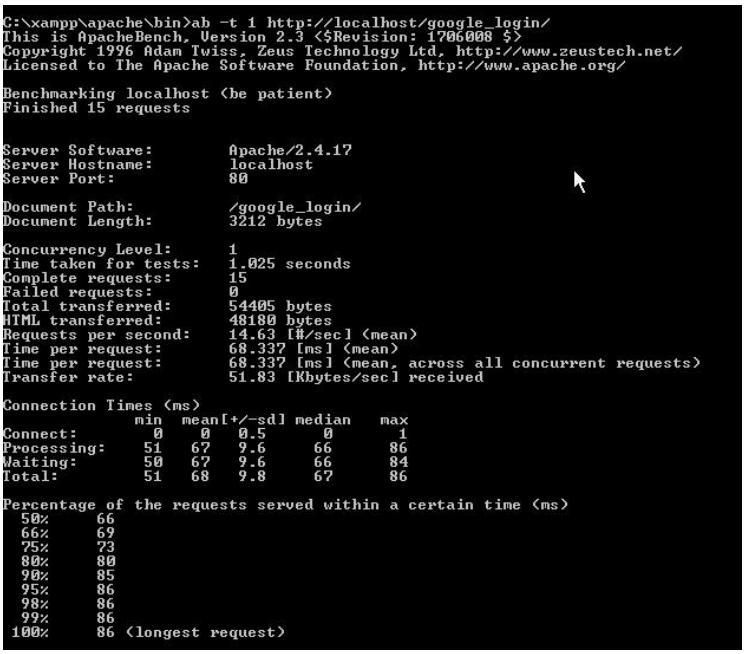
Tabel 1

Pengujian ke	Selang Waktu (s)	Rata-rata request CAS	Rata-rata request OAuth
1	1	471	15
2	2	1343	31
3	3	2149	47
4	4	2797	62
5	5	3568	76

Dari hasil test performa dapat dilihat bahwa sistem login CAS mampu menangani *request* jauh lebih banyak dari sistem login OAuth. Namun semua hasil yang didapat juga tergantung oleh perangkat-perangkat yang menjadi pendukung untuk sistem login. Capture dari test performa sistem login CAS dan OAuth dapat dilihat pada Gambar 16 dan 17.



Gambar 16. Hasil test response time dari cas



Gmabar 17. Hasil test response time dari oauth

3.2 PEMBAHASAN

Pada tahap analisa kebutuhan dan pengumpulan data, sumber atau referensi yang diacu adalah segala informasi yang berhubungan dengan penelitian, dimana referensi tersebut meliputi jurnal, thesis, skripsi, tutorial dan bahkan thread. Sedikitnya tutorial yang mengacu pada sistem login CAS mengakibatkan pembuatan sistem login CAS mengalami kendala, salah satunya adalah database pada CAS untuk diterapkan pada server tomcat. Sehingga penulis

sengaja menggunakan sistem default yang ada pada apikasi sistem login CAS. Sedangkan untuk sistem login OAuth, dengan sedikitnya jurnal yang mengangkat tentang OAuth tidak membuat penelitian terhambat. Karena thread, tutorial, serta pengembang yang mengangkat topik terkait sistem login OAuth untuk SSO sangatlah banyak.

Dalam masa penelitian login CAS dan OAuth, war file dari CAS yang masih default akan membuat TGT(Ticket Granting Ticket) pada saat user akan login. Dimana TGT adalah file identifikasi yang dibatasi oleh periode yang valid. Setelah authentication, file ini akan diberikan kepada pengguna untuk memproteksi jalur data. Didalam TGT terdapat session key dimana session key tersebut memiliki tanggal kadaluarsa tersendiri setelah diberikan kepada pengguna. Tomcat menjadi dasar yang cukup untuk menjalankan CAS yang berformat .war, dimana pada Tomcat terdapat catalina.bat yang berguna untuk melihat TGT yang dibuat saat CAS digunakan untuk login.

OAuth (Open Authorization) dengan menggunakan partisi pihak ketiga yaitu google, proses login menjadi lebih mudah dikarenakan pengguna cukup menggunakan akun google-nya untuk dapat login ke web yang telah terintegrasi dengan google. Apache server sebagai dasar untuk pembuatan sistem login OAuth, dimana Client ID, Client Secret juga URIs path akan diatur didalamnya.

Pada test performa sistem login antara CAS dan OAuth, tool yang digunakan adalah *Apache Bench* (AB), dimana test yang dilakukan adalah “Response Time” untuk melihat besar *request* yang mampu ditangani antara CAS dan OAuth. Dari hasil 5 kali percobaan per selang waktu yang didapat, diambil rata-rata untuk dijadikan sebagai data sample. Sebagai contoh percobaan untuk sistem login CAS dengan selang waktu 1 detik didapat 5 data request yang berbeda: 413 412 496 421 617. Dari data yang ada akan diambil rata-rata dan dijadikan sebagai data hasil percobaan.

4 PENUTUP

4.1 KESIMPULAN

Dari penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan antara lain:

1. CAS lebih susah untuk di terapkan sebagai SSO dalam web login daripada OAuth, karena kerumitan sistem CAS dan kurang populernya sistem ini serta sedikitnya tutorial yang mendukung.
2. OAuth lebih mudah di terapkan dalam SSO, selain karena mudahnya konfigurasi dan pengembangan OAuth, juga karena dalam perkembangannya OAuth jauh lebih banyak digunakan oleh kebanyakan developer. Banyaknya tutorial dan minat developer untuk

mengembangkan sistem login OAuth ini semakin memudahkan dalam pembuatan sistem login OAuth.

3. Dalam kemampuan menerima *request*, CAS jauh lebih baik dibandingkan OAuth, dimana semakin besar waktu tunda maka semakin besar jumlah *request* yang dapat ditangani.

4.2 SARAN

Untuk setiap developer yang ingin mengembangkan atau menggunakan sistem login dengan CAS atau OAuth sebagai dasar SSO, akan lebih mudah menggunakan OAuth untuk membangun sistem login yang menggunakan partisi pihak ketiga. Untuk CAS, dibutuhkan pemahaman yang lebih karena kerumitan yang ada pada sistem CAS dan kecocokan dari tools pendukung juga mempengaruhi.

DAFTAR PUSTAKA

Aminudin. (2014). Implementation of Single Sign On (SSO) Support For E-Commerce Interactivity Applications Using Protocol Oauth. *GAMMA*, Vol 10, No 1 (2014).

Amarudin. (2014). Implementation of CAS Server as Authentication Protocol on Single Sign-On(SSO) Network With PHP Programming. *ICETIA*. Informatics Engineering Department STMIK Teknokrat.

Grag, Parul. (2016). SSO (Single Sign On) Implementation. *International Journal of Sicence and Research (IJSR),* Volume 5 Issue 6, June 2016.

Muni. (2016). Login with Google OAuth 2 Using PHP and MySQL. Retrieved July 15, 2016, from http://www.smarttutorials.net/login-with-google-oauth-2-using-php-and-mysql.

Ramadhan, Gilang. (2013). Analisis teknologi Single Sign On (SSO) dengan penerapan Central Authentication Service (CAS) pada Universitas Bina Darma. *Bina Darma e-Journal*, Vol. xx No.x Oktober 2013: 1-13.

Kurniawan, Fite., Fajar Suryawan., & Umi Fadlilah. (2014). Membangun Privileges Pada Jaringan Komputer Sma Negeri 2 Boyolali Berbasis Active Directory Dengan Windows Server 2008 Enterprise. *Emitor*, Volume 14 No. 1, Maret 2014.

Al-Fedaghi, Sabah. Developing Web Applications. *International Journal of Software Engineering and Its Applications*, Vol.5 No.2, April, 2011